ARMY RESEARCH LABORATORY

Word Count

by George W. Hartwig, Jr.

ARL-MR-544 June 2002

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-MR-544 **June 2002**

Word Count

George W. Hartwig, Jr.
Computational and Information Sciences Directorate, ARL

Abstract

This report describes a computer program that takes text files and generates word frequency counts. It attempts to intelligently handle punctuation. It will also ignore words appearing on an exclusion list. The output may be sent to a file or will appear on the display. Setting the debug flag may produce annoying amounts of output if a large file is being processed. A word-stemming algorithm is optionally available if desired.

Acknowledgments

I would like to acknowledge Maria Lopez and Fred Brundick for taking the time review this document. I would also like to acknowledge LTC Robert Hammel, who provided the inspiration for this entire project.

INTENTIONALLY LEFT BLANK.

Table of Contents

		Page
	Acknowledgments	iii
	List of Figures	vii
	List of Tables	vii
1.	Introduction	1
2.	Methodology	1
2.1	Preprocessing	2
2.2	Word Extraction	2
2.3	Optional Stemming	3
2.4	Word Frequency Count	6
2.5	Optional Test Against Exclusion List	7
2.6	Print Results to File or Display	7
3.	Usage	8
4.	Results	8
5.	Conclusions	10
6.	References	11
	List of Abbreviations	13
	Distribution List	15
	Report Documentation Page	17

INTENTIONALLY LEFT BLANK.

List of Figures

Figure		Page
1.	Overview of Word Count Process	2
2.	Hyphen Removal	3
3.	The Frequency Count Process	6

List of Tables

Table		Page
1.	Typical Exclusion List	7
2.	First 20 Results With No Options	8
3.	First 20 Results With No Capitalization	9
4.	First 20 Results With No Caps and Exclusion List Applied	9
5.	First 20 Results With No Caps, Exclusion List, and Stemming	10

INTENTIONALLY LEFT BLANK.

1. Introduction

To make use of the opportunities offered by modern information warfare an effective means of selecting the crucial information from the profusion of digital data available on the battlefield is required. While at the higher echelons this task is performed by the command staff, at the lower echelons there is often no one to help the commander with this task. In fact, if a procedure for selectively presenting important information to the user is not devised, a very real threat exists that the intended recipient will miss this crucial data. To address this problem, the University of South Florida (USF) is evaluating the feasibility of applying techniques of soft computing, i.e., fuzzy logic, to prioritize input for the decision-maker's attention.

With this technique, word documents of interest are clustered into classes of interest. Word frequency counts are used as clustering vectors by an extended semi-supervised fuzzy c–means algorithm (ssFCM) [1]. Since this was to be a cooperative project, it was agreed that the U.S. Army Research Laboratory (ARL) would provide the word counting software for this project. An internet search was made to look for previously written software that met the project requirements including the ability to selectively perform word stemming, exclude common words,* and either ignore case or not. Several sites were located where word counting software could be found, but none met the requirements of the project. Two of the sites [2, 3] used word counting as an application to compare various languages, and none of the software found at these sites addressed the additional problems of word stemming or exclusion of common words. A third site [4] described a program that met all the requirements but was written in an obscure language called SPITBALL-386. Therefore, it was decided to create a new program using in-house resources to meet the specific project requirements including word stemming and the exclusion of user-specified words.

In this report, we will discuss the issues to be considered when creating a word-frequency counting program and how those issues are resolved. A straightforward algorithm to perform word stemming created by Porter [5] is also discussed.

2. Methodology

The general technique implemented by this program is shown in Figure 1.

As Johnson points out in his web page [4], there are a number of difficulties in deciding what are words and what are not words. These center around deciding what may be included in a word and what should not be included. Issues to be resolved include hyphens, apostrophes, digits, and non-English characters. These and other problems are discussed in the following paragraphs.

*These are commonly referred to as stop words and include words like "the," "and," etc. Several such lists may be found on the World Wide Web by searching with the key words "stop words."

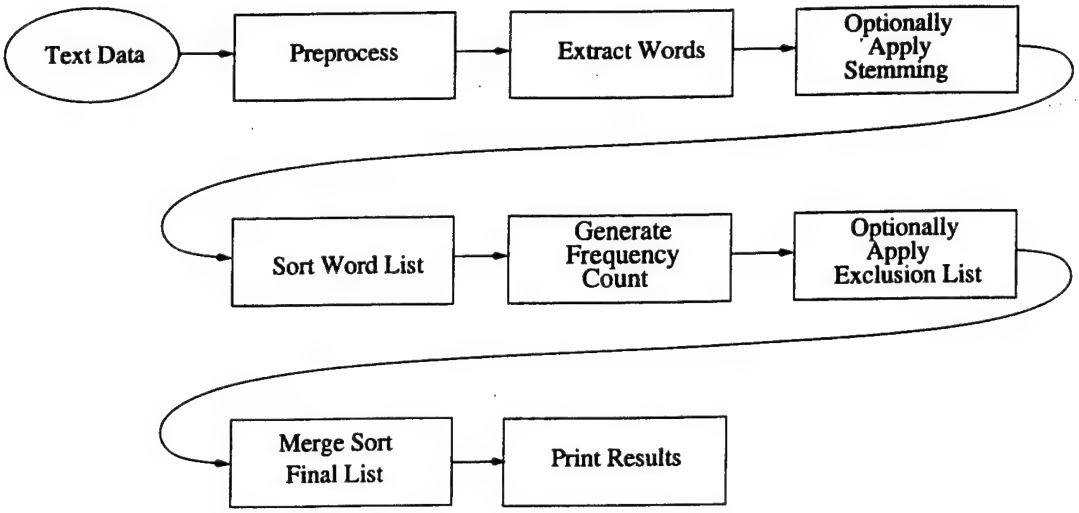

Figure 1. Overview of Word Count Process

2.1 Preprocessing

This module takes our raw text file and performs some elementary processing to ease the parsing into words that occurs later. First, if case is to be ignored, all characters are translated into lowercase. Then all punctuation is removed with a few exceptions: (1) Apostrophes within a word we will keep. Those either preceded or followed by a space character will be deleted. By space we mean any white space character such as the tab or the space character. (2) Embedded periods are retained; others are deleted unless preceded or followed by a numeric character. (3) A colon surrounded by digits may be a time, so we retain it, at least initially. (4) Finally we consider hyphens. An embedded hyphen will always be kept. One surrounded by white space will always be deleted. If a hyphen is followed by a newline,* both the hyphen and the newline are to be thrown out. The first word on the next line will be moved up two characters and the resulting slots filled with spaces. This process is shown in Figure 2.

2.2 Word Extraction

Once the preprocessing is completed, a word list is made consisting of all recognizable words in the file. Since much of the punctuation was removed in the preprocessing stage, we recognize the space, carriage return, LF, tab, and null or end-of-file characters as word delimiters. Processing is done in approximately 1000 character blocks. It is approximate because if a word spans the 1000 character boundary, the entire word is included in the block. The words found are stored in an array that is dynamically allocated and increased in size as needed.

*Newline characters are treated differently in UNIX operating systems and Windows. In UNIX the newline character is represented in a file by the line feed (LF) character. Windows, on the other hand, uses a carriage return and LF. So this program would need to be modified to run correctly on Windows systems.

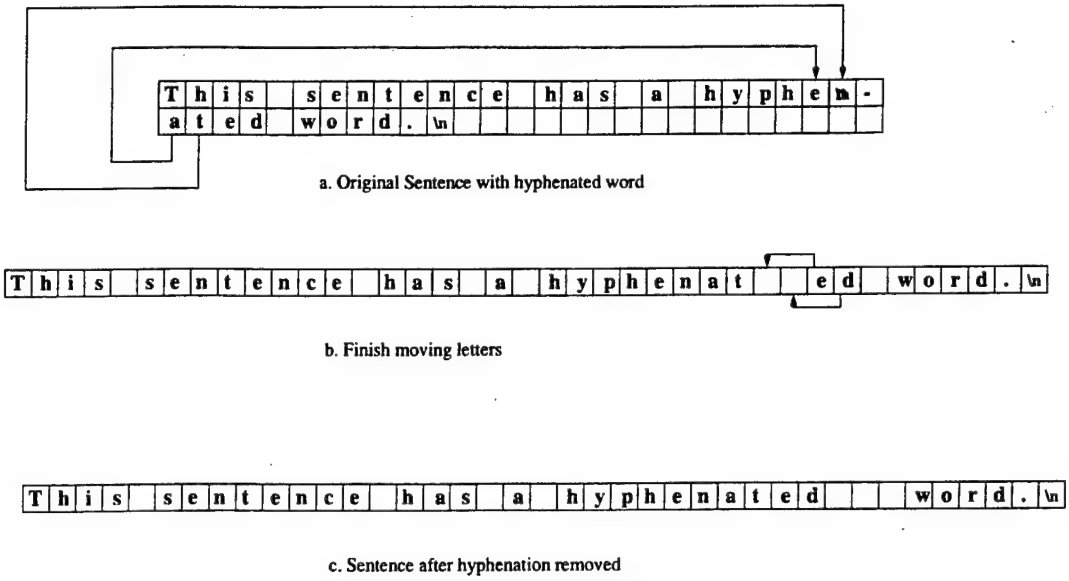

Figure 2. Hyphen Removal

2.3 Optional Stemming

Word stemming is simply reducing each word considered to its root. This reduces the final word list and simplifies searching for particular words. For instance,

connecting
connected
connection

all have the same root–"connect." To perform the stemming operation an algorithm by Martin F. Porter was selected [6]. While no practical stemming algorithm is perfect, the Porter algorithm is a good compromise between computability and precision. An implementation written by Porter [5] in the C programming language was used in this work.

The Porter algorithm operates by defining a list of suffixes, and for each suffix a set of rules describe the conditions under which the suffix can be removed. To understand the algorithm, we must first know some of Porter's terminology. His abstract definition of a word is

$$[\mathbf{C}](\mathbf{VC})m[\mathbf{V}],$$

where

C represents a sequence of one or more consonants,
V represents a sequence of one or more vowels,
[ ] mean that the contents may or may not be present, and
$(\mathbf{VC})m$ denotes **VC** repeated m times. m may be zero.

Rules for suffix removal are of the form

(condition) S1 → S2,

where

S1 is the original suffix, and
S2 is the suffix that is to replace it.

The condition is often an expression involving m and may include the following terms:

*S – The stem ends with the letter S or any other specified letter.
v – The stem contains a vowel.
*d – The stem ends with a double consonant.
*o – The stem ends **cvc**, where the second consonant is not W, X, or Y.

Expressions may be combined with the boolean operators "and," "or," and "not."

What follows are the rules implemented for this program. They are grouped in several sections and only a single rule from each section is executed. The evaluation of some sections depend on certain rules being executed in the previous section. Within a section, rules are listed below each other. For a given word, only the longest match on the left-hand side is executed. Except for the terms previously defined, all letters are literal and case insensitive. For a more detailed explanation of these rules and examples, see [6].

Section 1a

SSES → SS
IES → I
SS → SS
S →

Section 1b

(m>0) EED → EE
(*v*) ED →
(*v*) ING →

If the second or third rule in step 1b is executed, then the following rules are checked.

AT → ATE
BL → BLE
IZ → IZE
(*d and not (*L or *S or *Z)) → single letter
(m =1 and *o) → E

Section 1c

(*v*) Y → I

Section 2

(m>0) ATIONAL → ATE
(m>0) TIONAL → TION
(m>0) ENCI → ENCE
(m>0) ANCI → ANCE
(m>0) IZER → IZE
(m>0) ABLI → ABLE
(m>0) ALLI → AL
(m>0) ENTLI → ENT
(m>0) ELI → E
(m>0) OUSLI → OUS
(m>0) IZATION → IZE
(m>0) ATION → ATE
(m>0) ATOR → ATE
(m>0) ALISM → AL
(m>0) IVENESS → IVE
(m>0) FULNESS → FUL
(m>0) OUSNESS → OUS
(m>0) ALITI → AL
(m>0) IVITI → IVE
(m>0) BILITI → BLE

Section 3

(m>0) ICATE → IC
(m>0) ATIVE →
(m>0) ALIZE → AL
(m>0) ICITI → IC
(m>0) ICAL → IC
(m>0) FUL →
(m>0) NESS →

Section 4

(m>1) AL →
(m>1) ANCE →
(m>1) ENCE →
(m>1) ER →
(m>1) IC →
(m>1) ABLE →
(m>1) IBLE →
(m>1) ANT →
(m>1) EMENT →
(m>1) MENT →
(m>1) ENT →
(m>1 and (*S or *T) ION →
(m>1) OU →
(m>1) ISM →

(m>1) ATE →
(m>1) ITI →
(m>1) OUS →
(m>1) IVE →
(m>1) IZE →

Step 5a

(m>1) E →
(m=1 and not *o) E →

(m>1 and *d and *L) → single letter

2.4 Word Frequency Count

After the preprocessing is completed, the text stream is parsed into words, and these words are sorted alphabetically. This sorting facilitates the frequency count, aids in subsequent processing, and results in the list of words and counts being in alphabetic order.

Figure 3 shows the process used in creating the frequency count. The first word is taken from the word list, stored in a word frequency array, and the count associated with this word is incremented. Then the next word is taken from the word list and compared to the previous word. If they are the same, the count is incremented, and the process continues. If they are not the same, the new word is stored in the word frequency array and its count incremented. This process continues until the word list is exhausted.

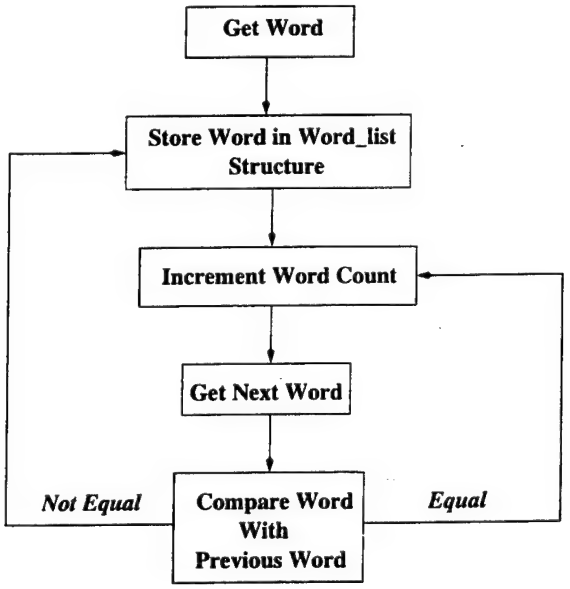

Figure 3. The Frequency Count Process

2.5 Optional Test Against Exclusion List

Often when counting the frequency with which words are found in a document, it is desirable to be able to exclude some words such as articles, pronouns, some verbs, and other common words. Shown in Table 1 are typical words found in an exclusion list.* This list is by no means all inclusive, and it is expected that each application of the word counting program would require additions and, perhaps, deletions from this list.

Table 1. Typical Exclusion List

the	all	even	words	of	this	were	still
and	one	some	sight	to	he	other	nothing
a	no	how	upon	i	your	him	most
in	so	any	time	that	our	who	know
is	more	then	own	you	an	them	himself
my	their	than	way	it	on	being	same
as	had	been	also	by	would	once	whole
not	if	into	those	but	what	very	its
for	they	such	myself	be	we	should	let
with	line	shall	thus	or	will	point	themselves
at	now	out	while	his	has	do	because
which	when	before	thy	me	there	these	whose
was	can	nor	itself	have	could	call	whom
from	must	many	are	thou			

Given this exclusion list, the program searchs the list of words. When a word that is on the exclusion list is found, the corresponding count is set to zero. The final list is resorted by count, from highest to lowest. Since the list was sorted alphabetically before being tested against the exclusion list, a merge sort [8] is used to move all those words with zero counts to the end of the word list where they are ignored during any further processing.

2.6 Print Results to File or Display

This program prints the resulting words and their associated frequency counts to either the display or a specified file. Note that if debugging is turned on, output can be voluminous.

*This exclusion list was derived by running the word frequency program on an electronic book [7]. After the frequency counts were generated, a manual selection was made to choose the words that met the criteria previously mentioned.

3. Usage

Use of the word counting program is straightforward and is shown below:

```
USAGE: wc [-d -c -l -s -u config_file] -i infile -o outfile -e excl_file
   -d              = Execute debugging prints (can produce a lot of output).
   -c config_file  = Contains parameters to modify operation of program.
                     NOT IMPLEMENTED.
   -i infile       = The file containing the data to be analyzed.
   -e exclude_file = The file containing words to be excluded from analysis.
   -l              = Convert all alphabetic characters to lower case.
   -s              = print word stems only (Porter algorithm).
   -o outfile      = Put all the output into a file.
   -u/-h           = Print this message.
```

The configure file is not yet implemented but is included to support future growth. All options are specified on the command line. If no outfile is provided, the resulting word list is written to the display. If debugging prints are enabled, the resulting output is also sent to the display.

4. Results

A digital copy of the United States Constitution with its 27 amendments was used for demonstration [9]. A run with none of the options invoked found 8118 words of which 1486 were unique. The first 20 are shown in Table 2.

Table 2. First 20 Results With No Options

Rank	Word	Frequency	Rank	Word	Frequency
1.	the	700	11.	President	108
2.	of	520	12.	a	96
3.	shall	305	13.	United	86
4.	and	263	14.	for	83
5.	to	188	15.	State	78
6.	be	180	16.	any	78
7.	or	161	17.	Congress	76
8.	in	141	18.	The	66
9.	by	128	19.	as	64
10.	States	122	20.	have	64

As we can see, this list of words is predominantly articles, verbs, and adjectives.

Table 3 shows the first 20 words from a run with the "-l" option, converting all capital letters to lowercase. This process reduces the number of unique words to 1295. The count for "the" is now 769. This is the result of the words "the," "The," and "THE" being combined into the single word "the."

Table 3. First 20 Results With No Capitalization

Rank	Word	Frequency	Rank	Word	Frequency
1.	the	769	11.	president	108
2.	of	520	12.	a	102
3.	shall	305	13.	united	86
4.	and	269	14.	for	84
5.	to	207	15.	state	80
6.	be	180	16.	any	78
7.	or	161	17.	congress	76
8.	in	150	18.	section	68
9.	by	129	19.	as	65
10.	states	127	20.	have	64

Table 4 gives the results when all capitals are converted to lowercase and words in the exclusion list are deleted, command line options "-l -e exclusionList". No longer are the results dominated by articles, pronouns, verbs, etc. The number of unique words is further reduced to 1208.

Table 4. First 20 Results With No Caps and Exclusion List Applied

Rank	Word	Frequency	Rank	Word	Frequency
1.	states	127	11.	office	37
2.	president	108	12.	house	33
3.	united	86	13.	person	33
4.	state	80	14.	article	30
5.	congress	76	15.	representatives	29
6.	section	68	16.	senate	28
7.	amendment	49	17.	each	26
8.	may	46	18.	vice	26
9.	law	39	19.	number	25
10.	constitution	37	20.	ratified	25

Finally, Table 5 shows the top results when the Porter Stemming Algorithm is applied to the previous results, command line options are "-l -s -e exclusionList". The number of unique words is reduced to 978 words. Notice that some of the words have been significantly mangled. For instance, "president" has become "presid" and "united" has become

"unit". Changes such as these make it imperative that users know the effects of applying the algorithm when requiring the stemming algorithm to be applied.

Table 5. First 20 Results With No Caps, Exclusion List, and Stemming

Rank	Word	Frequency	Rank	Word	Frequency
1.	state	209	11.	mai	46
2.	presid	109	12.	constitut	43
3.	unit	86	13.	hous	41
4.	congress	76	14.	vote	37
5.	section	69	15.	power	35
6.	offic	57	16.	repres	34
7.	amend	56	17.	year	33
8.	law	52	18.	articl	31
9.	senat	49	19.	number	29
10.	person	48	20.	each	26

5. Conclusions

This report has described a program for producing word frequency counts from computer text files. It allows the optional inclusion of the Porter Word Stemming algorithm. It is being successfully used to preprocess the text files being clustered by the ssFCM algorithm. Since this program was created to support work being performed at USF in accordance with a contract with ARL, no further development of the program is contemplated. The program remains a viable candidate for those problems where word frequency counts are required.

6. References

1. Bensaid, A., L. Hall, J. Bezdek and L.P. Clarke. "Partially Supervised Clustering for Image Segmentation." *Pattern Recognition*, vol. 29, no. 5, pp. 859–871, 1996.
2. Bouchard, J. "Counting Words in a Text Comparison of Several Algorithms and Languages." <http://mageos.ifrance.com/bouchard/>, May 2000.
3. Bagley, D. "Word Frequency." <http://www.bagley.org/ doug/shootout/bench/wordfreq>, August 2001.
4. Johnson, E. "Counting Words and Computing Word Frequency." <http://www.dsu.edu/ johnsone/words.html>, November 2000.
5. Porter, M. F. "An Algorithm for Suffix Stripping." <http://www.tartarus.org/ martin/PorterStemmer/def.txt>, May 1980.
6. Porter, M. F. "C Implementation of the Porter Stemming Algorithm." <http://www.tartarus.org/ martin/PorterStemmer/c.txt>.
7. Abbot, E. A. *Flatland*, <http://wiretap.area.com/Gopher/Library/Classic/flatland.txt>.
8. Gasch, S. "A Mergesort Implemented in C." <http://wannabe.guru.org/alg/node53.html>, July 1999.
9. National Archives and Records Administration. "United States Constitution." <http://www.nara.gov/exhall/charters/constitution/constitution.html>, July 1996.

INTENTIONALLY LEFT BLANK.

List of Abbreviations

ARL	U.S. Army Research Laboratory
USF	University of South Florida
ssFCM	Semi-supervised Fuzzy C Means

INTENTIONALLY LEFT BLANK.

NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFORMATION CENTER DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	HQDA DAMO FDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	OSD OUSD(A&T)/ODDR&E(R) DR R J TREW 3800 DEFENSE PENTAGON WASHINGTON DC 20301-3800
1	COMMANDING GENERAL US ARMY MATERIEL CMD AMCRDA TF 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN 3925 W BRAKER LN STE 400 AUSTIN TX 78759-5316
1	US MILITARY ACADEMY MATH SCI CTR EXCELLENCE MADN MATH THAYER HALL WEST POINT NY 10996-1786
1	DIRECTOR US ARMY RESEARCH LAB AMSRL D DR D SMITH 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB AMSRL CI AI R 2800 POWDER MILL RD ADELPHI MD 20783-1197

NO. OF COPIES	ORGANIZATION
3	DIRECTOR US ARMY RESEARCH LAB AMSRL CI LL 2800 POWDER MILL RD ADELPHI MD 20783-1197
3	DIRECTOR US ARMY RESEARCH LAB AMSRL CI IS T 2800 POWDER MILL RD ADELPHI MD 20783-1197
	ABERDEEN PROVING GROUND
2	DIR USARL AMSRL CI LP (BLDG 305)

NO. OF
COPIES ORGANIZATION

1 UNIV OF S FLORIDA
DEPT OF CMPTR SCI ENGRNG
L HALL
4202 E FOWLER AVE
TAMPA FL 33620

ABERDEEN PROVING GROUND

7 DIR USARL
AMSRL CI CT
J BRAND
A BRODEEN
F BRUNDICK
S CHAMBERLAIN
G HARTWIG
P JONES
M LOPEZ

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2002	Final, November 2000–May 2001

4. TITLE AND SUBTITLE
Word Count

5. FUNDING NUMBERS
611102.AH48
2TEPMC

6. AUTHOR(S)
George W. Hartwig, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-CI-CT
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-MR-544

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes a computer program that takes text files and generates word frequency counts. It attempts to intelligently handle punctuation. It will also ignore words appearing on an exclusion list. The output may be sent to a file or will appear on the display. Setting the debug flag may produce annoying amounts of output if a large file is being processed. A word-stemming algorithm is optionally available if desired.

14. SUBJECT TERMS
word frequency counts, word stemming, soft computing, C language

15. NUMBER OF PAGES
20

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.